Date=4/12/2020
Lecture By=Manish Mahant
Subject ⇒ Project Day-3

IN PREVIOUS LECTURE **(QUICK RECAP) Date-3/12/2020**	**In Today's Lecture (Overview)**
Login With Google Using ReactJS Create ReactJS Project Add Routing in ReactJS Add Google Sign-In and Sign-Out Button and Get the AccessToken	Redux Authentication Reducer connect() Overview

Redux Authentication Reducer

Path: /src/_reducers/authentication.reducer.js

The redux authentication reducer manages the state related to login (and logout) actions, on successful login the current user object and a loggedIn flag are stored in the `authentication` section of the application state. On

logout or login failure the authentication state is set to an empty object, and during login (between login request and success/failure) the authentication state has a loggingIn flag set to true and a user object with the details of the user that is attempting to login.

```
import { userConstants } from '../_constants';


let user = JSON.parse(localStorage.getItem('user'));
const initialState = user ? { loggedIn: true, user } : {};


export function authentication(state = initialState, action) {
    switch (action.type) {
    case userConstants.LOGIN_REQUEST:
        return {
        loggingIn: true,
        user: action.user
        };
    case userConstants.LOGIN_SUCCESS:
        return {
        loggedIn: true,
        user: action.user
        };
    case userConstants.LOGIN_FAILURE:
        return {};
```

```
    case userConstants.LOGOUT:

        return {};

    default:

        return state

    }

}
```

connect()

Overview

The connect() function connects a React component to a Redux store.
It provides its connected component with the pieces of the data it needs from the store, and the functions it can use to dispatch actions to the store.
It does not modify the component class passed to it; instead, it returns a new, connected component class that wraps the component you passed in.

```
function connect(mapStateToProps?, mapDispatchToProps?, mergeProps?,
options?)
```

The mapStateToProps and mapDispatchToProps deals with your Redux store's state and dispatch, respectively. state and dispatch will be supplied to your mapStateToProps or mapDispatchToProps functions as the first argument.
The returns of mapStateToProps and mapDispatchToProps are referred to internally as stateProps and dispatchProps, respectively. They will be supplied to mergeProps, if defined, as the first and the second argument, where the third argument will be ownProps. The combined result, commonly referred to as mergedProps, will then be supplied to your connected component.

For Complete Project

https://github.com/attainu/aryabhata-course-module/tree/master/m-tube

Previous Notes Regarding This Project

Day1

https://docs.google.com/document/d/1wkK18Ka6CBKLgA25JKrGaEPmUX20QBA6SrkqMsDmQe0/edit?usp=sharing

Day2
https://docs.google.com/document/d/1uq-v6ZUXdkK3bJmvbBscR6623eOORPU39dJY_abwMH0/edit?usp=sharing